Continuous Database Monitoring with the Trace API

Firebird Conference 2011 – Luxembourg

Thomas Steinmaurer DI

+43 7236 3343 896
thomas.steinmaurer@scch.at
t.steinmaurer@upscene.com
www.scch.at www.upscene.com
http://blog.upscene.com/thomas/
Motivation
Monitoring Approaches
Firebird 2.5 Audit and Trace Services
Usage of fbtracemgr
FB TraceManager 2
Q&A
Motivation

- Several stakeholders for database monitoring
  - Administrators
  - Developers
  - Auditors
  - ...

- A snapshot (MON$) of current activities is interesting. But, most of the time you need the history of executed events

- Continuous monitoring gives you a stream of executed events over a period of time and lets you look back

- A new way to detect and diagnose problems

- The **Audit and Trace Services API in Firebird 2.5** provides that!!!
Typical use cases for the **Audit and Trace Services API**

- Number of executed statements for a particular time period
  - Usage statistics for load planning
- Block-box debugging
  - Execution trace for an in-house / third-party product
  - For third-party products, usually you don‘t have access to the client source code
  - No need for adding trace logic into the client application
- Detection of
  - Failed / Unauthorized access
  - Certain event types
  - Improper client transaction management (e.g. Regular usage of COMMIT RETAINING)
  - Full table scans
**Motivation**

- Typical use cases for the **Audit and Trace Services API**
  - Detection of (continued)
    - Most-Frequently executed statements
    - Worst-Performing executed statements
    - Executed statements which are slower than a given threshold (e.g. 5 seconds)
  - Statistics
    - Number of statements per IP / process name / user ...
    - Most loaded database on the server
    - When are usual peak loads
    - ...
  - Generated trace as input for a security audit
  - **AND MORE ... !!!**
Quick Start Demonstration
Agenda

- Motivation
- Monitoring Approaches
- Firebird 2.5 Audit and Trace Services
- Usage of fbtracemgr
- FB TraceManager 2
- Q&A
Basic monitoring requirements
- Continuous
- Server-side
- Configurable

Three out-of-the-box monitoring approaches in Firebird 2.5
- Trigger-based (since Firebird 1.0)
- Monitoring tables (since Firebird 2.1)
- Audit and Trace Services (since Firebird 2.5)

Based on your monitoring requirements, you possibly will have/use a mix of these approaches
## Monitoring Approaches Comparison

<table>
<thead>
<tr>
<th></th>
<th>Trigger-based</th>
<th>Monitoring tables</th>
<th>Audit/Trace Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available since Firebird version</td>
<td>1.0</td>
<td>2.1</td>
<td>2.5</td>
</tr>
<tr>
<td>Continuous</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Serverside</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Configurable</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Monitoring of SELECT</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Monitoring of DELETE, INSERT, UPDATE</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
# Monitoring Approaches Comparison

<table>
<thead>
<tr>
<th>Access to old/new column values for DELETE, INSERT, UPDATE</th>
<th>Trigger-based</th>
<th>Monitoring tables</th>
<th>Audit/Trace Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Access to statement execution time</th>
<th>No</th>
<th>No</th>
<th>Yes</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Access to statement execution plan</th>
<th>No</th>
<th>No</th>
<th>Yes</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Monitoring of database-wide operations like connect, disconnect, transaction start etc.</th>
<th>Trigger-based</th>
<th>Monitoring tables</th>
<th>Audit/Trace Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partly (database triggers in Firebird 2.1)</td>
<td>Partly</td>
<td>Partly</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Agenda

- Motivation
- Monitoring Approaches
- **Firebird 2.5 Audit and Trace Services**
- Usage of fbtracemgr
- FB TraceManager 2
- Q&A
### Audit and Trace Services

**General**

- New feature in Firebird 2.5 across all architectures and supported platforms
- Allows to produce a trace of events in chronological sequence
- Event types
  - Database-specific
    - Connect, Disconnect
    - Start/Commit/Rollback of transactions
    - Prepare/Start/Finish/Free of SQL statements
    - Start/Finish of stored procedures and triggers
    - etc.
  - Server-wide
    - Services API requests (e.g. backup, restore, get server log etc.)
Audit and Trace Services
System Audit vs. User Trace

- **System Audit**
  - Will be started by the engine upon Firebird server start
  - Stores trace output in a text file on the server
  - Is activated by setting the new `AuditTraceConfigFile` parameter (= path to the trace configuration file) in `firebird.conf`
  - There can be only one system audit per Firebird instance

- **User Trace**
  - Needs to be started by a user manually
  - Trace output is read by the initiating application fetching trace data via the Services API
  - Needs to be re-started after a server crash/restart
  - There can be several user traces per Firebird instance
Implemented by the pre-installed fbtrace(.dll|.so) plugin

Trace session ID 1 started
2010-11-10T22:26:00.4160 (1368:0000000000F1DC88) TRACE_INIT
  SESSION_1 FBTM - tourism

2010-11-10T22:26:00.4160 (1368:0000000000F1DC88) ATTACH_DATABASE
tourism.fdb (ATT_1415, TOURISM:NONE, NONE, TCPv4:127.0.0.1)
  C:\Program Files (x86)\Upscene Productions\Database Workbench 4
  Pro\DBW4.exe:3768

2010-11-10T22:26:00.4220 (1368:0000000000F1DC88) START_TRANSACTION
tourism.fdb (ATT_1415, TOURISM:NONE, NONE, TCPv4:127.0.0.1)
  C:\Program Files (x86)\Upscene Productions\Database Workbench 4
  Pro\DBW4.exe:3768
  (TRA_24395, READ_COMMITTED | REC_VERSION | NOWAIT | READ_WRITE)

2010-11-10T22:26:00.4860 (1368:0000000000F1DC88) PREPARE_STATEMENT
tourism.fdb (ATT_1415, TOURISM:NONE, NONE, TCPv4:127.0.0.1)
  C:\Program Files (x86)\Upscene Productions\Database Workbench 4
  Pro\DBW4.exe:3768
  (TRA_24395, READ_COMMITTED | REC_VERSION | NOWAIT | READ_WRITE)
Audit and Trace Services
Trace Configuration

- A system audit / user trace is configurable by providing a mandatory trace configuration (file)

- Customization of
  - Database-wide events
    - Connect, disconnect, prepare statement etc.
  - Server-wide events
    - Services API requests

- Consists of
  - Default <database> section
  - <database db_name_pattern> section per database
  - Max. one <services> section for server-wide events

- Pre-installed fbtrace.conf is a good start for writing your own trace configuration
**A very simple example**

```xml
<database>
    enabled true
</database>

<database employee.fdb>
    enabled false
    log_connections true
    log_transactions true
</database>

<database tourism.fdb>
    log_connections true
</database>
```
Audit and Trace Services
Trace Configuration

- **<database>** configuration parameters
  - enabled, log_filename, max_log_size, include_filter, exclude_filter, log_connections, connection_id, log_transactions, log_statement_prepare, log_statement_free, log_statement_start, log_statement_finish, log_procedure_start, log_procedure_finish, log_trigger_start, log_trigger_finish, log_context, print_plan, print_perf, log_blr_requests, print_blr, log_dyn_requests, print_dyn, time_threshold, max_sql_length, max_blr_length, max_dyn_length, max_arg_length, max_arg_count

- **<services>** configuration parameters
  - enabled, log_filename, max_log_size, include_filter, exclude_filter, log_services, log_service_query
Trace Sessions can be managed via new Services API calls

<table>
<thead>
<tr>
<th>Action</th>
<th>Services API call</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start a new user trace session</td>
<td>isc_action_svc_trace_start</td>
</tr>
<tr>
<td>Stop a trace session</td>
<td>isc_action_svc_trace_stop</td>
</tr>
<tr>
<td>Suspend a trace session</td>
<td>isc_action_svc_trace_suspend</td>
</tr>
<tr>
<td>Resume a trace session</td>
<td>isc_action_svc_trace_resume</td>
</tr>
<tr>
<td>List all trace sessions</td>
<td>isc_action_svc_trace_list</td>
</tr>
</tbody>
</table>
Audit and Trace Services
Security

- Every user can start a trace session
- Used user name upon attaching to the services manager defines
  - Privileges in respect to managing other trace sessions
  - Visibility of traceable statements
- Privileges
  - SYSDBA can manage all other trace sessions including a system audit
  - Non-SYSDBA users can manage their own trace sessions only
- Visibility
  - A user trace session started by SYSDBA logs activities of all users/connections
  - A user trace session started by a non-SYSDBA user logs only its own activities
Agenda

- Motivation
- Monitoring Approaches
- Firebird 2.5 Audit and Trace Services
- Usage of fbtracemgr
- FB TraceManager 2
- Q&A
Usage of fbtracemgr
Live Demonstration
Agenda

- Motivation
- Monitoring Approaches
- Firebird 2.5 Audit and Trace Services
- Usage of fbtracemgr
- FB TraceManager 2
- Q&A
Live Demonstration
FB TraceManager 2
Editions

- **Lite Edition**
  - Local connections and raw trace data only
  - Freely available

- **Standard Edition**
  - Lite + Remote connections, Parser, Reporting
  - **99 EUR** per user named license

- **Enterprise Edition**
  - Standard + Analysis, Event Processing
  - **149 EUR** per user named license

- **Trial Edition**
  - 30 days limited Enterprise Edition
  - Limited number of received/parsed trace events
Agenda

- Motivation
- Monitoring Approaches
- Firebird 2.5 Audit and Trace Services
- Usage of fbtracemgr
- FB TraceManager 2
- Q&A
Questions and Answers
Thanks for your attention!

thomas.steinmaurer@scch.at

t.steinmaurer@upscene.com

http://blog.upscene.com/thomas
Resources

- Trace Services and Audit in Firebird 2.5, by Vlad Khorsun; Firebird Conference 2009 Munich
- FB TraceManager 2: http://www.upscene.com/go/?go=fbtm